DPD-DFF: A Dual Phase Distributed Scheme with Double Fingerprint Fusion for Fast and Accurate Identification in Large Databases

Daniel Peralta[a,*], Isaac Triguero[b,c], Salvador García[a,d], Francisco Herrera[a], Jose M. Benitez[a]

[a]*Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain*
[b]*Department of Respiratory Medicine, Ghent University, 9000 Gent, Belgium*
[c]*VIB Inflammation Research Center, 9052 Zwijnaarde, Belgium*
[d]*Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

Abstract

Nowadays, many companies and institutions need fast and reliable identification systems that are able to deal with very large databases. Fingerprints are among the most used biometric traits for identification. In the current literature there are fingerprint matching algorithms that are focused on efficiency, whilst others are based on accuracy.

In this paper we propose a flexible dual phase identification method, called DPD-DFF, that combines two fingers and two matchers within a hybrid fusion scheme to obtain both fast and accurate results. Different alternatives are designed to find a trade-off between runtime and accuracy that can be further tuned with a single parameter.

The experiments show that DPD-DFF obtains very competitive results in comparison with the state-of-the-art score fusion techniques, especially when dealing with large databases or impostor fingerprints.

Keywords: Real-time identification, large databases, minutiae matching, fingerprint fusion, decision fusion, score fusion, parallel computing, biometrics

1. Introduction

Personal identification has arisen as an important issue in the last few years for many companies and institutions [1]. Identification databases grow larger every year, ranging from a few tens of people for small companies to several millions for institutions such as the police. Although there are various biometric traits that allow for identification, fingerprints are widely used because of their uniqueness and universality, among other properties [2, 3]. Fingerprint recognition can be tackled from two different perspectives: verification [4] and identification [5]. The former consists of matching two fingerprints to determine whether they belong to the same finger or not. The latter aims to identify an input fingerprint from a set of fingerprints and determine which of them matches with the input. In

*Corresponding author. Tel.: +34 958244019; fax: +34 958243317
Email addresses: dperalta@decsai.ugr.es (Daniel Peralta), Isaac.Triguero@irc.vib-UGent.be (Isaac Triguero), salvagl@decsai.ugr.es (Salvador García), herrera@decsai.ugr.es (Francisco Herrera), J.M.Benitez@decsai.ugr.es (Jose M. Benitez)

this context, an Automatic Fingerprint Identification System (AFIS) is a tool that allows us to perform identifications in fingerprint databases [3].

Fingerprints are composed of a pattern of ridges and valleys, from which diverse features can be extracted. Among these features, minutiae are widely used for fingerprint matching, mostly due to their distinctiveness [2, 6]. When two fingerprints are to be compared, the minutiae are extracted from the images, and then a matching algorithm is applied over the two minutiae sets to determine a similarity level. There are multiple proposals of minutiae-based matching algorithms in the literature [7]. Some of them are very efficient due to their simplicity [8], while others are very accurate [9]. However, these two objectives are usually not reached together because accurate algorithms tend to be complex, and therefore time-consuming. This restriction complicates the development of AFIS that are able to identify people in very large databases in a suitable time frame without precision loss.

Moreover, as the overall response time of an identification procedure is linear with respect to the size of the database, even the fastest matching algorithms may become useless when the database grows too large. Moreover, the huge number of matchings causes an accuracy loss.

Information fusion is a widely used paradigm that improves overall precision in many fields, including biometrics [10, 11, 12]. In particular, two main approaches have been proven to enhance the recognition capabilities: the use of several fingerprint images [13] and the use of several matching algorithms [14]. The information fusion can be performed at different levels:

- Feature fusion approaches merge the characteristics extracted from different fingerprint images, coming either from the same finger or different fingers [15, 16].
- Score fusion methods perform separate matchings and then sum up the scores [14, 17].
- Decision fusion methods apply the matching algorithms in a hierarchical mode over the fingerprints [11, 18].

Although these approaches increase the accuracy of the AFIS, they also slow the identification down because the processing workload is higher. In this work, we combine the ideas of multi-finger and multi-algorithm identification to improve the runtime along with the accuracy.

High Performance Computing (HPC) is an important tool to speed up the runtime of a system [19, 20], and several proposals in the literature apply it to AFIS. However, these systems focus on objectives other than precision, such as high availability [21], load balancing [22] or reduced matching times [18]. Other systems provide the ability to identify in very large databases [23, 24, 25], but their accuracy is not improved with respect to a sequential AFIS.

There are currently several systems in the world that maintain large fingerprint databases. For example, as of September 2015, India's UIDAI system [26] stores the fingerprints of around 907 million people, although so far they are only used for verification purposes, not identification. FBI IAFIS [27] (now included within Next Generation Identification, NGI) keeps the fingerprints (among other data) for around 104 million subjects, and is able to perform searches in an average time of 72 minutes.

In this paper, we propose a flexible, Dual Phase Distributed AFIS with Double Fingerprint Fusion (called DPD-DFF) that integrates two fingerprints and two matching algorithms, aiming to overcome the weaknesses of isolated approaches: high identification time and accuracy loss. To do so, the identification is split into two phases, each of which can either use a single fingerprint or fuse two of them, conforming a mixed score fusion and decision fusion process:

- In the first phase, the database is explored by a fast matching algorithm to select a candidate set. Jiang's algorithm [8] has been selected for this phase due to its high running speed [7].
- Then, the second phase applies a more accurate algorithm to identify the correct identity within this candidate set. The matcher used in this phase is Minutia Cylinder-Code (MCC) [9], which is very precise [7].

With this design, the fingerprint fusion is powerful and flexible as it is performed at two separate levels. Furthermore, this strategy has been integrated within the parallel framework proposed in [23] in order to reach full scalability for arbitrarily large databases.

This manuscript is structured as follows. First, Section 2 provides the background information on the problem at hand. Section 3 presents DPD-DFF, the approach proposed in this paper. Section 4 describes the experiments performed and their results. Finally, Section 5 details the conclusions. Complementary material to the paper including tables, plots and identification times as well as additional studies over other databases can be found at `http://sci2s.ugr.es/DPDDFF` and in the associated Technical Report [28].

2. Preliminaries

A fingerprint is a pattern of valleys and ridges located on a fingertip. Although there are several ways to perform a matching between two fingerprints, many matching algorithms use the minutiae [3, 7, 29], comparing two minutiae sets to return a similarity score. The matching is performed once for each comparison between two fingerprints. Some of the existing matching algorithms offer very good matching precision [9], and others provide a fast response with slightly diminished accuracy [8], according to the taxonomy and results presented in [7].

There are two main variants of the fingerprint recognition problem [3]. Verification [4] is a 1:1 comparison to check if two fingerprints represent the same finger. Identification [5] consists of determining which fingerprint in a database of previously captured and stored templates $T = \{T_1, T_2, ..., T_n\}$ corresponds to a given input fingerprint I. An identification algorithm compares I to every T_i and returns the identity with the best matching score as shown in Eq. 1, where $Q(I, T_i)$ is the matching function. Thus, identification is a 1:n comparison.

$$\text{Identity} = \arg\max_i Q(I, T_i) \quad i \in \{1, 2, ..., n\} \tag{1}$$

This paper is focused on identification. Section 2.1 explains the current proposals for fast and scalable identification within large databases. Then, Section 2.2 presents the previous work about fingerprint fusion to improve the identification accuracy.

2.1. Scalable fingerprint recognition in large databases

The bottleneck of an AFIS when attempting to identify within a large database is the matching algorithm. Several proposals in the literature aim to overcome this problem.

FPGA-based systems implement the matching into a Field Programmable Gate Array [18, 30], a hardware device that performs some operations very quickly, so that the overall identification time is reduced.

Other approaches reduce the penetration rate in the database by using a previous classification or indexing step [31, 32, 33, 34]. Nevertheless, in large databases this step may become the bottleneck, and the size of the subsets can become too large. Accuracy is degraded when the penetration rate is too small or the collision rate too high [33].

HPC is a common solution for reducing high execution times [19, 20]. By using q computers with c cores each to perform a parallel search, the execution time can be reduced by up to a factor of qc. Moreover, the availability of more RAM memory allows more template fingerprints to be kept in a fast access device, avoiding slow access to secondary memory. Therefore, an adequate parallel framework can constitute a suitable tool for solving the identification problem in large databases [23, 24, 25].

2.2. Fingerprint information fusion

This section introduces two of the main trends to improve the accuracy of fingerprint recognition. On the one hand, the use of several fingers [13] increases the distinctiveness of the identities and tries to avoid the difficulties posed by injured fingertips or low quality scans. The matching function for f fingerprints becomes of the form $Q(\mathcal{I}, \mathcal{T}_i)$ where $\mathcal{I} = \{I_j \mid j \in \{1, ..., f\}\}$ and $\mathcal{T}_i = \{T_{ij} \mid j \in \{1, ..., f\}\}$. This approach has been successfully applied over latent fingerprints, which are of very low quality [35].

On the other hand, the combination of several matchers [14, 36] aims to profit from their advantages, while leaving aside their weaknesses. Multi-algorithm techniques work in a similar way as multi-finger ones, so that the fused score obtained for f algorithms is $Q(I, T_i) = \mathcal{F}\left(Q_1(I, T_i), ..., Q_f(I, T_i)\right)$, where $\mathcal{F}$ is an aggregation function.

Multi-finger and multi-algorithm approaches can be categorized together according to the type of fusion they perform:

- **Feature fusion** [15, 16, 37, 38]: this approach merges all f fingerprints of an identity into a single structure, which is compared to all n template structures. This avoids the necessity of performing f matchings per identity, but requires specific matching algorithms to handle such structures, as well as an additional conversion step.
- **Score fusion** [14, 17, 36, 39, 40]: this method applies several matchings (one for each fingerprint or algorithm) and aggregates the results into a single score. Although it does not need a specific matching algorithm, the use of f fingerprints or f matchings multiplies the identification time by f.

- **Decision fusion** [10, 11, 18, 32]: can be seen as a special case of score fusion, where matching is performed hierarchically. When the f input fingerprints are compared with some f template fingerprints for a given identity, the first pair is compared first. If the resulting score meets a certain condition, the second pair is compared, and so on.

Most fusion approaches are focused on improving accuracy, without considering runtime. Therefore, they are not adapted to address the identification in large databases because the execution time is higher than it is for simpler approaches. Empirical results obtained by some of the methods mentioned above can be found in the Technical Report associated with this paper [28].

3. Dual Phase Distributed Scheme with Double Fingerprint Fusion

DPD-DFF carries out a hybrid fusion between two matching algorithms and two fingers within a flexible dual phase scheme that is implemented in a parallel HPC system. The proposal seeks to tackle large fingerprint databases with a good trade-off between two seemingly opposed objectives:

- **Accuracy**: identification accuracy must be better than it is for isolated models.
- **Efficiency and scalability**: the system should provide a real-time response. The runtime threshold depends on the specific application; it can vary between a few milliseconds and several minutes. Ideally the identification time should be lower than when using an isolated AFIS.

First, a fast matcher explores the whole database and extracts a set of candidate identities C. Then, an accurate matcher compares the input fingerprints with the templates in C. This corresponds to a decision fusion identification method as described in Section 2.2, in which the separate use of both algorithms avoids the necessity of transforming their respective outputs to a common domain and the consequent loss of precision, as it does for traditional multi-algorithm score fusion approaches. The overall identification procedure is applied as follows:

1. **Fast phase**: according to the results obtained in [7], Jiang's algorithm [8] has been selected to perform this first identification phase, because of its speed and its appropriate accuracy. Two different criteria may be used to compose the set C:
   - **Rank**: given a rank r, select the r identities that provide the best scores. Thus, C has a fixed size $|C| = r$.
   - **Threshold**: all templates $\mathcal{T}_i$ whose score is higher than a fixed threshold ϕ when compared to the input fingerprint $\mathcal{I}$ are included in C. Therefore, the size of C is not previously known and will likely be different for each input fingerprint pair. The set can be described as $C = \{\mathcal{T}_i \mid Q_{Jiang}(\mathcal{I}, \mathcal{T}_i) \geq \phi\}$.
2. **Accurate phase**: the MCC algorithm [9] has been chosen for this phase due to its high accuracy. After comparing the input fingerprints with the templates in C, the identity with the best score is returned as the found match, as shown in Eq. 2.

$$\text{Identity} = \arg\max_i \{Q_{MCC}(\mathcal{I}, \mathcal{T}_i) \mid \mathcal{T}_i \in C\} \tag{2}$$

$$T_{AB} = \{\mathcal{T}_i \mid \mathcal{T}_i = \{T_{iA}, T_{iB}\}\ ,\ i \in \{1, 2, ..., n\}\} \tag{3}$$

In addition to this multi-algorithm scheme, we also use two different fingers (let them be finger A and finger B) per identity to even further improve identification accuracy. Two template fingerprints per person are stored, constituting a database T_{AB} with n fingerprints pairs as described in Eq. 3. An identification requires an input set of two fingerprints $\mathcal{I} = \{I_A, I_B\}$. According to this structure, each of the previously described identification phases can be carried out using either a single fingerprint or both fingerprints:

- **Single finger**: a single fingerprint of each identity is compared, as shown in Eq. 4. This alternative is proposed in a search for speed, minimizing the computation load.

$$\begin{aligned} Q_{\text{Jiang}}(\mathcal{I}, \mathcal{T}_i) &= Q_{\text{Jiang}}(I_A, T_{iA}) \quad \text{(fast phase)} \\ Q_{\text{MCC}}(\mathcal{I}, \mathcal{T}_i) &= Q_{\text{MCC}}(I_B, T_{iB}) \quad \text{(accurate phase)} \end{aligned} \tag{4}$$

- **Double finger**: both fingerprints are used for the comparison. This constitutes in itself a fusion method. Thus, a score-based fusion has been implemented, using the average as the aggregation function (Eq. 5), as recommended by the results of [17]. This approach is obviously slower than using a single finger, but it is much more accurate.

$$\begin{aligned} Q_{\text{Jiang}}(\mathcal{I}, \mathcal{T}_i) &= \frac{Q_{\text{Jiang}}(I_A, T_{iA}) + Q_{\text{Jiang}}(I_B, T_{iB})}{2} \quad \text{(fast phase)} \\ Q_{\text{MCC}}(\mathcal{I}, \mathcal{T}_i) &= \frac{Q_{\text{MCC}}(I_A, T_{iA}) + Q_{\text{MCC}}(I_B, T_{iB})}{2} \quad \text{(accurate phase)} \end{aligned} \tag{5}$$

Table 1: Names of the eight considered variants of DPD-DFF

Fingers used			Candidate set criterion		
First phase	Second phase	Prefix	Rank (*R)	Threshold (*T)	Objective
Single (A)	Single (B)	SS*	**SSR**	**SST**	High speed
Single (A)	Double (A,B)	SD*	**SDR**	**SDT**	Trade-off
Double (A,B)	Single (B)	DS*	**DSR**	**DST**	Trade-off
Double (A,B)	Double (A,B)	DD*	**DDR**	**DDT**	High accuracy

The described method performs a hybrid fusion that uses both score and decision fusion to combine two fingers and two algorithms. The overall workflow is depicted in Figure 1. A pseudocode of the identification procedure

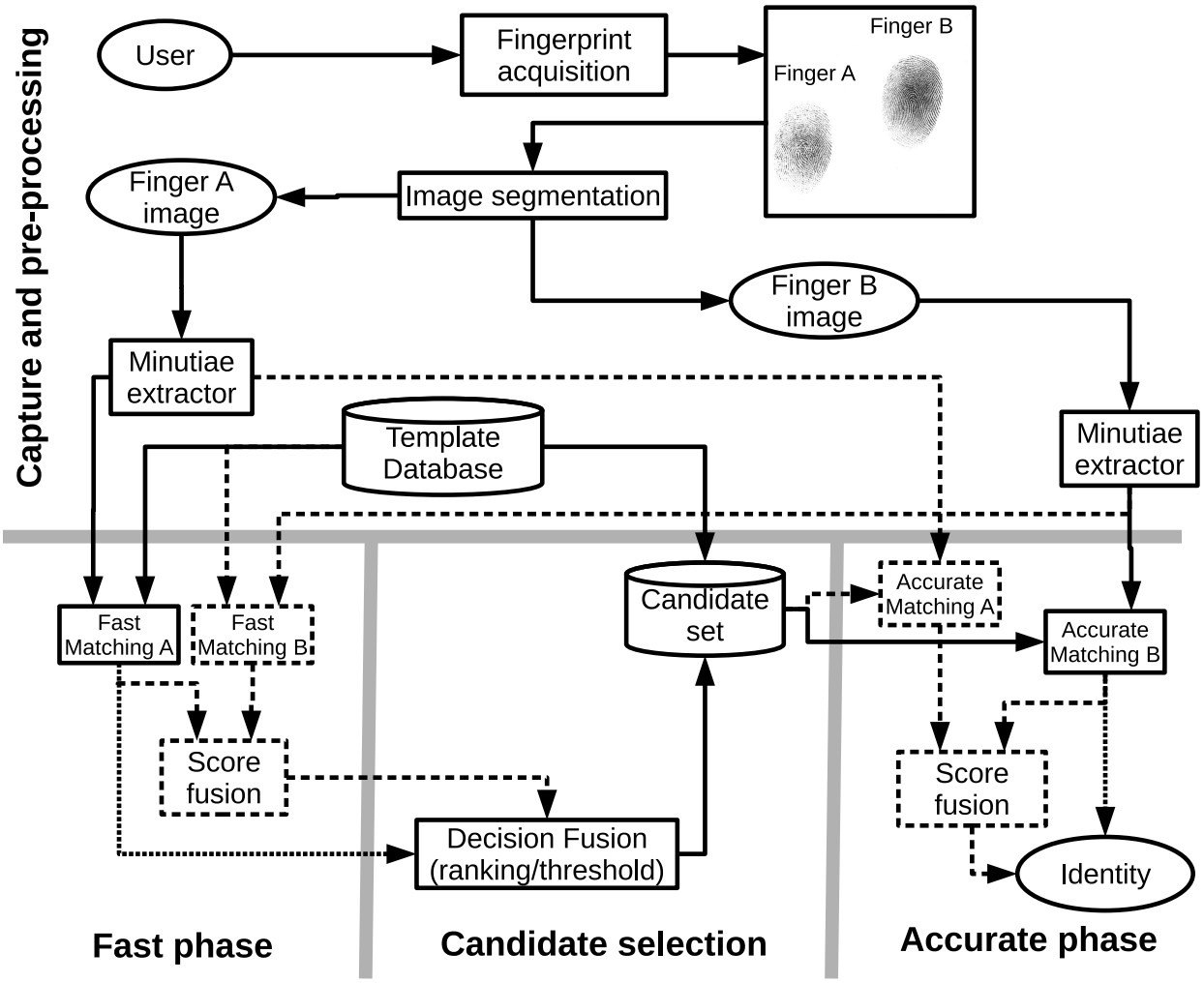


Figure 1: Workflow of DPD-DFF. Dashed lines are pathways that correspond to double finger variants. Dotted lines correspond to single finger variants. Continuous lines are pathways that are always taken.

is shown in Algorithm 1. Out of this design, we take eight variants of the algorithm into consideration, which are denoted with three letters as shown in Table 1. The first two letters represent the fingers that are taken for the fast and accurate phases respectively (S for single and D for double). The last letter stands for the criterion to build the candidate set (R for rank, T for threshold).

Note that the variants that use both fingers within a same phase (SD*, DS* and DD*) will eventually apply both matching algorithms over the fingerprints in C. This can enhance the identification accuracy, due to the synergy between two algorithms that perform the matching differently [14].

Along with the algorithm variant, the choice of the parameters to build the candidate set (r or θ) is critical, as it will determine its size $|C|$, which in turn determines the trade-off between speed and accuracy: a small candidate set relies more on the fast phase and provides faster results (though less accurate), whilst a large candidate set leads to

Input: T, $\mathcal{I}$, crit, r, θ

$C \leftarrow \emptyset$

`// Fast phase`

foreach $\mathcal{T}_i \in T$ **do**

  $q \leftarrow Q_{Jiang}(\mathcal{I}, \mathcal{T}_i)$

  if *crit* == *"Ranking"* **then**

    if $|C| < r$ **then** C.append($\mathcal{T}_i$);

    else

      $min_C = \arg\min_i \{Q_{Jiang}(\mathcal{I}, \mathcal{T}_i) \mid \mathcal{T}_i \in C\}$

      if $Q_{Jiang}(\mathcal{I}, \mathcal{T}_{min_C}) < q$ **then**

        C.remove($\mathcal{T}_{min_C}$)

        C.append($\mathcal{T}_i$)

      end

    end

  else if *crit* == *"Threshold"* ***and*** $Q_{Jiang}(\mathcal{I}, \mathcal{T}_i) \geq \theta$ **then**

    C.append($\mathcal{T}_i$)

  end

end

`// Accurate phase`

$max_q \leftarrow 0$

$identity \leftarrow null$

foreach $\mathcal{T}_i \in C$ **do**

  if $Q_{MCC}(\mathcal{I}, \mathcal{T}_i) > max_q$ **then**

    $max_q \leftarrow Q_{MCC}(\mathcal{I}, \mathcal{T}_i)$

    $identity \leftarrow \mathcal{T}_i$

  end

end

return *identity*

Algorithm 1: DPD-DFF algorithm

more accurate results but needs longer runtime.

Despite the separation between fast and accurate phases, if the structure proposed so far is implemented in a sequential manner the scalability problem will eventually appear for arbitrarily large databases. To achieve high scalability, DPD-DFF has been developed within the two-level parallel framework proposed in [23], as described in the Technical Report [28]. Hence, the scheme can be efficiently executed in a cluster of computers.

4. Experiments and results

This section describes the experiments performed over several fingerprint databases: a large database of SFinGe-generated fingerprints (Section 4.2), a database captured by the authors (Section 4.3), the well-known NIST-DB14 (Section 4.4) and several other public databases (Section 4.5). Section 4.1 describes the hardware and software used for these experiments.

The number of True Positives (TP), False Positives (FP) and False Negatives (FN) are used as accuracy measures, along with the True Positive Rate (TPR). The average identification time is denoted by t_{avg} and measured in seconds in all cases. For the threshold variants, the average candidate set size $|C|_{avg}$ is also given. The plots include the accuracy and identification time of three reference AFIS: an isolated one that uses a single finger and a single matcher (as described in [8, 9]), and two score fusion approaches, one multi-finger (as described in [17, 41]) and one multi-algorithm (as described in [11, 17, 36]).

Note that for a fair comparison, both DPD-DFF and the reference AFIS were implemented in the framework proposed in [23] and executed over the same hardware. It is out of the scope of this paper to analyze the performance of the parallel procedure; the study is focused on the behavior of the proposed hybrid fusion method.

Additional details and results (such as tables, figures, database statistics, identification times, hardware configuration and results with more databases) are available in the associated Technical Report [28] and at `http://sci2s.ugr.es/DPDDFF`.

4.1. Hardware and software environment

The experiments carried out for this paper have been executed in a cluster of 12 nodes, each of them with two Intel Xeon E5-2620 processors (6 cores each). The executions were performed with 12 slave processes (one in each node), each of them composed of 24 threads. Note that a smaller subset of nodes was used for the databases of small size.

All fingerprint minutiae were extracted using the NIGOS *mindtct* software [42], whose parameters are detailed in Table 2. The authors have written their own implementation of the underlying matching algorithms [8, 9], with the sole aid of their respective original publications. The parameters used for these algorithms are also presented in the table.

To ensure a fair comparison, the same parameters were used for all the tested databases, so as to avoid any kind of over-fitting of the results. Even though this may produce low accuracy values for some of the databases, this setup aims to assess the robustness of the proposed method in different use cases.

4.2. SFinGe database

This section describes the experiments performed over a database of 50 000 fingerprint pairs, synthetically generated with the SFinGe software [3, 43]. First, Section 4.2.1 details the used fingerprint database. Then, Section 4.2.2 describes the experiments carried out and the obtained results.

Table 2: Parameters for the methods used in the experimentation

Algorithm	Parameters	Reference
Jiang	$w_d = 1, w_\theta = 0.3\frac{180}{\pi}, w_\phi = 0.3\frac{180}{\pi}$ $w_n = 0, w_t = 0$, Consolidation step iterations = 5 Minutiae neighborhood size = 2 $BG_1 = 8, BG_2 = \frac{\pi}{6}, BG_3 = \frac{\pi}{6}$	[8]
MCC	$R = 70, N_s = 8, N_d = 6, \sigma_s = \frac{28}{3}, \sigma_d = \frac{2\pi}{9}$ $\mu_\Psi = 0.01, \tau_\Psi = 400, \omega = 50, min_{VC} = 0.75$ $min_M = 2, min_{ME} = 0.60, \sigma_\theta = \frac{\pi}{2}, max_{n_p} = 12$ Floating-point-based version: enabled, $\mu_P = 20$ $w_R = 0.5, \mu_1^\rho = 5, \tau_P = 0.6, min_{n_p} = 4, \tau_1^\rho = -1.6$ $\mu_2^\rho = \frac{\pi}{12}, \tau_2^\rho = -30, \mu_3^\rho = \frac{\pi}{12}, \tau_3^\rho = -30, n_{rel} = 5$	[9]
mindtct	output format = ANSI INCITS 378-2004 image enhancement = enabled	[42]

4.2.1. Database generation and parameters of the algorithms

In order to obtain very large databases and to control the fingerprint characteristics, we used the SFinGe software [3, 43] to generate synthetic fingerprints using the parameters specified in Table 3. The fingerprint pairs are composed by joining two synthetic fingerprints. A fingerprint cannot be included in more than one pair to ensure that all pairs are unique and disjoint in the database.

Table 3: Parameter specification used with the SFinGe tool

Scanner parameters	Generation parameters	Output settings
Acquisition area: 14.6mm x 19.6mm.	Impression per finger: 25.	Output file type: WSQ.
Resolution: 500 dpi.	Class distribution: Natural.	
Image size: 288 x 384.	Varying quality and perturbations.	
Background type: Optical.	Generate pores: enabled.	
Background noise: Default.	Save ISO templates: enabled.	
Crop borders: 0 x 0.		

The test set for all the experiments carried out in this paper with the SFinGe database is composed of 1000 random input pairs, which are used to perform 1000 different identifications in the database of 50 000 template fingerprint pairs. Each input pair is formed by a different impression of each fingerprint of a template pair. Therefore, we obtain accuracy measures that range from 0 to 1000.

4.2.2. Discussion of the results

This section discusses the obtained results for all the described variants of DPD-DFF over the SFinGe database. For the experiments, the rank values used to build the candidate set have been taken among the multiples of the number of cores of the cluster (144 in our setup), to maximize the throughput. However, we have also used lower values of the rank those in order to enrich the study and obtain more information about the behavior of the obtained accuracy.

Table 4: Results of DPD-DFF with 1000 test identifications (rank)

	SSR				**SDR**				**DSR**				**DDR**			
r	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)
12	927	73	0	0.1588	928	72	0	0.1845	994	6	0	0.2870	996	4	0	0.3174
24	948	52	0	0.1597	949	51	0	0.1872	994	6	0	0.2776	997	3	0	0.3190
48	956	44	0	0.1589	958	42	0	0.1884	993	7	0	0.2815	997	3	0	0.3199
96	968	32	0	0.1597	970	30	0	0.1882	993	7	0	0.2889	997	3	0	0.3190
144	972	28	0	0.1606	974	26	0	0.1889	995	5	0	0.2833	999	1	0	0.3212
288	973	27	0	0.1603	976	24	0	0.1890	995	5	0	0.2916	999	1	0	0.3230
576	980	20	0	0.1696	983	17	0	0.2095	994	6	0	0.2977	999	1	0	0.3447
1152	984	16	0	0.1889	988	12	0	0.2680	992	8	0	0.3215	999	1	0	0.3788
2304	990	10	0	0.2412	995	5	0	0.3345	993	7	0	0.3485	1000	0	0	0.4483
4608	989	11	0	0.2897	996	4	0	0.4547	990	10	0	0.4185	1000	0	0	0.5887
9216	987	13	0	0.4313	996	4	0	0.7369	990	10	0	0.5665	1000	0	0	0.8665
18432	988	12	0	0.7379	998	2	0	1.3333	990	10	0	0.8481	1000	0	0	1.4356
36864	989	11	0	1.4270	1000	0	0	2.5521	989	11	0	1.4728	1000	0	0	2.5772

Table 5: Results of DPD-DFF with 1000 test identifications (threshold)

	SST					**SDT**				**DST**					**DDT**			
ϕ	$\lvert C\rvert_{avg}$	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	$\lvert C\rvert_{avg}$	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)
0.05	46528.2	989	11	0	1.5153	1000	0	0	2.9382	49242.4	989	11	0	1.7302	1000	0	0	3.1920
0.10	22913.0	989	11	0	0.8381	999	1	0	1.6204	23580.8	990	10	0	1.0345	1000	0	0	1.8055
0.15	5404.7	989	11	0	0.3223	993	7	0	0.5450	2511.3	993	7	0	0.3727	999	1	0	0.4911
0.20	685.4	964	30	6	0.1925	967	27	6	0.2319	81.7	995	4	1	0.2865	997	2	1	0.3197
0.25	39.9	911	54	35	0.1583	912	53	35	0.1878	1.8	969	0	31	0.2819	969	0	31	0.3170
0.30	1.5	796	26	178	0.1515	796	26	178	0.1792	0.9	884	0	116	0.2722	884	0	116	0.3115

Tables 4 and 5 present the results of the eight variants of DPD-DFF. Note that columns $|C|_{avg}$ and t_{avg} contain average values over the 1000 performed identifications. Accordingly, Figure 2 plots the TPR along with the average identification time (note the logarithmic scale) for each variant of DPD-DFF and each reference AFIS. The following highlights can be extracted:

- The accuracy increases along with the amount of used information, so that DS* and DD* approaches are the most accurate ones.
- For a same average candidate set size, the rank approach produces more accurate results than the threshold variants, especially for small candidate sets. This might seem surprising because given an input pair, if both variants produce a candidate set of the same size, then these candidate sets are the same. However, recall that

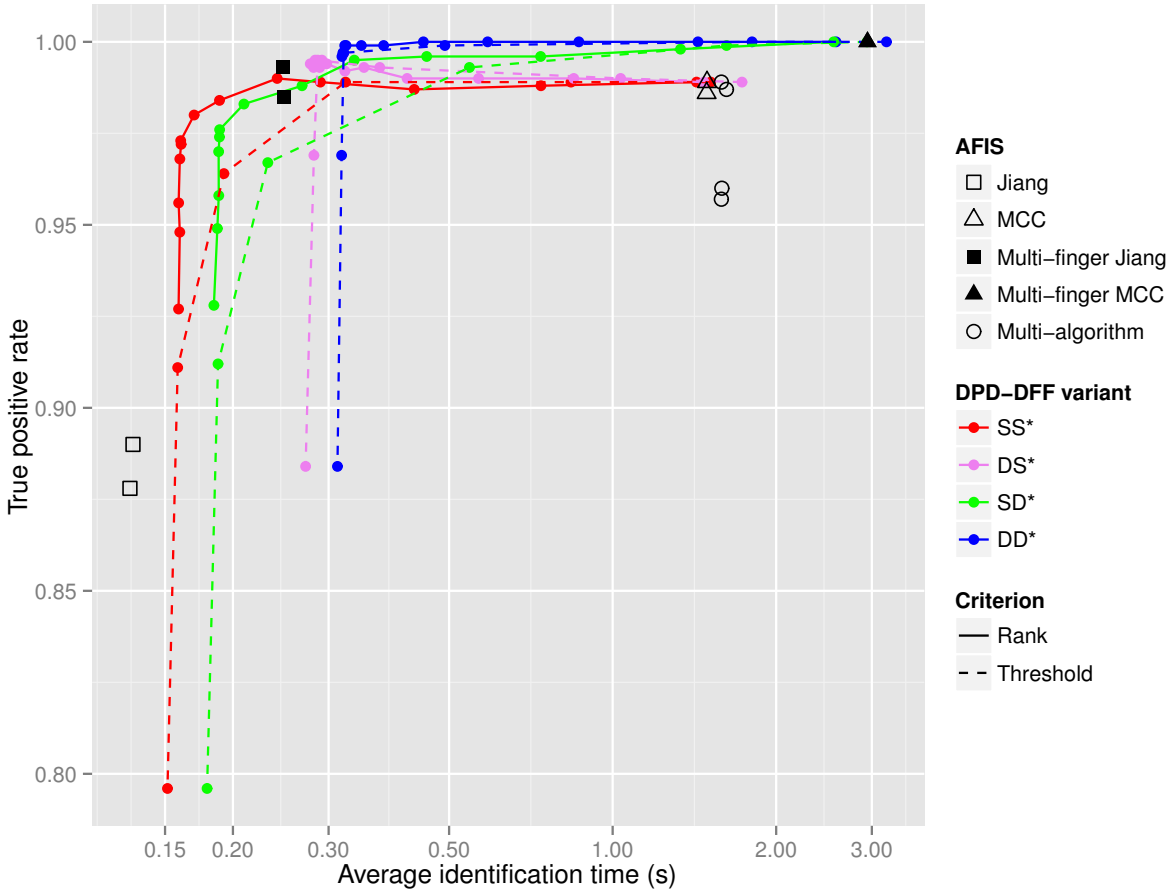


Figure 2: Average identification time and accuracy with the SFinGe database ($N = 50000$)

Table 5 shows the average set size. The actual value of $|C|$ in the threshold variant is different for each input pair, which makes the variant less robust.

- The rank ensures that there are no false negatives because it allows us to fix the size of the candidate set, ensuring $|C| > 0$ and offering better control of the overall identification time.
- Even for a small $|C|$, the rank variants outperform Jiang over a single fingerprint.
- Similarly, for a very large $|C|$, DPD-DFF relies more on the second phase and therefore the results get closer to those obtained by MCC. The DS* variants are a particular case because the accuracy decreases as $|C|$ increases. As the candidate set grows, they rely less on multi-finger Jiang, and more on single-finger MCC, which is less accurate than the former.
- The most accurate variant is DDR, which uses the two fingerprints with both algorithms and the rank.

- The DS* and DD* variants of DPD-DFF outperform all reference AFIS, reaching 100% TPR along with the multi-finger approach with MCC.

If the average identification time is also taken into account, the following conclusions arise:

- As expected, in general the more precise variants also take more time. These results show how DPD-DFF can be tuned according to the system needs, so that reasonably good results can be obtained very quickly (SSR variant, $r = 576$), and very precise results can be obtained with slower configurations (DDR variant, $r = 2304$).

- These tables also show that given a certain variant, the configurations with small candidate sets have a very similar runtime because all the matchings can be performed in parallel, but the accuracy is better for bigger candidate sets. Therefore, in a real environment, configurations with less than one candidate per core are usually not interesting, as they do not use the whole capacity of the cluster.

In summary, the DPD-DFF model dominates in time and accuracy all the tested AFIS, even the multi-finger approaches which provide very good accuracy. The DDR variant reaches 100% TPR in about 0.45 seconds, while the only reference AFIS that reaches this accuracy (multi-finger MCC) takes 3 seconds.

4.3. DBSpain654

A database of 654 fingerprint pairs was captured by the authors to test DPD-DFF on a controlled framework. This section describes the database (Section 4.3.1), the results obtained (Section 4.3.2), and an additional study with impostor fingerprints (Section 4.3.3). Due to the size of this database, all experiments described in this section were carried out using a single computer.

4.3.1. Database description

The fingerprints belong to the forefinger and middle finger of both hands of 334 non-experienced subjects from three different cities. Note that 14 fingerprint pairs failed in their enrolment and therefore were excluded from the database, making the resulting number of 654 pairs.

Both fingerprints of each pair were captured within the same image using a Suprema RealScan-D sensor. Each pair was captured 2 times as a template and 12 as an input over 3 different sessions several weeks apart. To compose the database and the test input set for this study, a single template and a single random input capture were selected for each pair. Then, the NIGOS *nfseg* algorithm [42] was used to segment the image and separate both fingerprints of each pair before applying the minutiae extraction.

4.3.2. Discussion of the results

Tables 6 and 7 present the results of the eight variants of the proposed DPD-DFF. Figure 3 depicts both accuracy and the average identification time of all tested AFIS. The following conclusions can be extracted from these results:

- The algorithms behave in the same way as in the previously studied databases: Jiang is less precise than MCC, and the multi-finger approaches obtain the best results both among the reference AFIS and the DPD-DFF variants.
- Again, the rank variants show more robust behavior than the threshold ones for the same average size of the candidate set. The DDR variant obtains the best performance possible for any number of candidates.
- DDR and DSR dominate all the considered multi-algorithm AFIS, and get the same TPR as multi-finger MCC in a much faster time.
- The multi-finger Jiang algorithm is faster than DPD-DFF. Actually, it corresponds to the first phase of the DS* and DD* variants, and it is clear that its accuracy is significantly improved with a small time overhead.

Table 6: Results of DPD-DFF with 654 test identifications (rank)

	SSR				**SDR**				**DSR**				**DDR**			
r	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)
2	614	40	0	0.0622	614	40	0	0.1000	652	2	0	0.0891	653	1	0	0.1264
4	623	31	0	0.0641	623	31	0	0.1020	652	2	0	0.0902	654	0	0	0.1277
8	629	25	0	0.0655	629	25	0	0.1030	651	3	0	0.0915	654	0	0	0.1296
12	634	20	0	0.0658	635	19	0	0.1040	648	6	0	0.0919	654	0	0	0.1302
24	639	15	0	0.0666	642	12	0	0.1058	647	7	0	0.0927	654	0	0	0.1315
48	642	12	0	0.0805	645	9	0	0.1330	647	7	0	0.1073	654	0	0	0.1598

Table 7: Results of DPD-DFF with 654 test identifications (threshold)

	SST					**SDT**				**DST**					**DDT**			
ϕ	$\lvert C\rvert_{avg}$	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	$\lvert C\rvert_{avg}$	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)
0.15	113.5	641	13	0	0.1165	646	8	0	0.2114	61.1	647	7	0	0.1205	654	0	0	0.1863
0.20	19.3	631	17	6	0.0694	632	16	6	0.1128	4.1	645	1	8	0.0893	646	0	8	0.1266
0.25	2.0	600	19	35	0.0619	600	19	35	0.0972	1.0	616	2	36	0.0867	616	2	36	0.1211
0.30	0.8	545	2	107	0.0590	545	2	107	0.0924	0.9	562	0	92	0.0857	562	0	92	0.1194

4.3.3. Results using impostor fingerprints

This section provides additional accuracy results for the DBSpain654 database.

In this section, the introduction of impostor fingerprints in the database requires additional error measures to study the behavior of DPD-DFF:

- **False Acceptance Rate (FAR)**: rate of impostor fingerprints that are erroneously identified as genuine ones.
- **False Rejection Rate (FRR)**: rate of genuine fingerprints that are erroneously rejected.
- **Equal Error Rate (EER)**: error when FAR and FRR are equal.

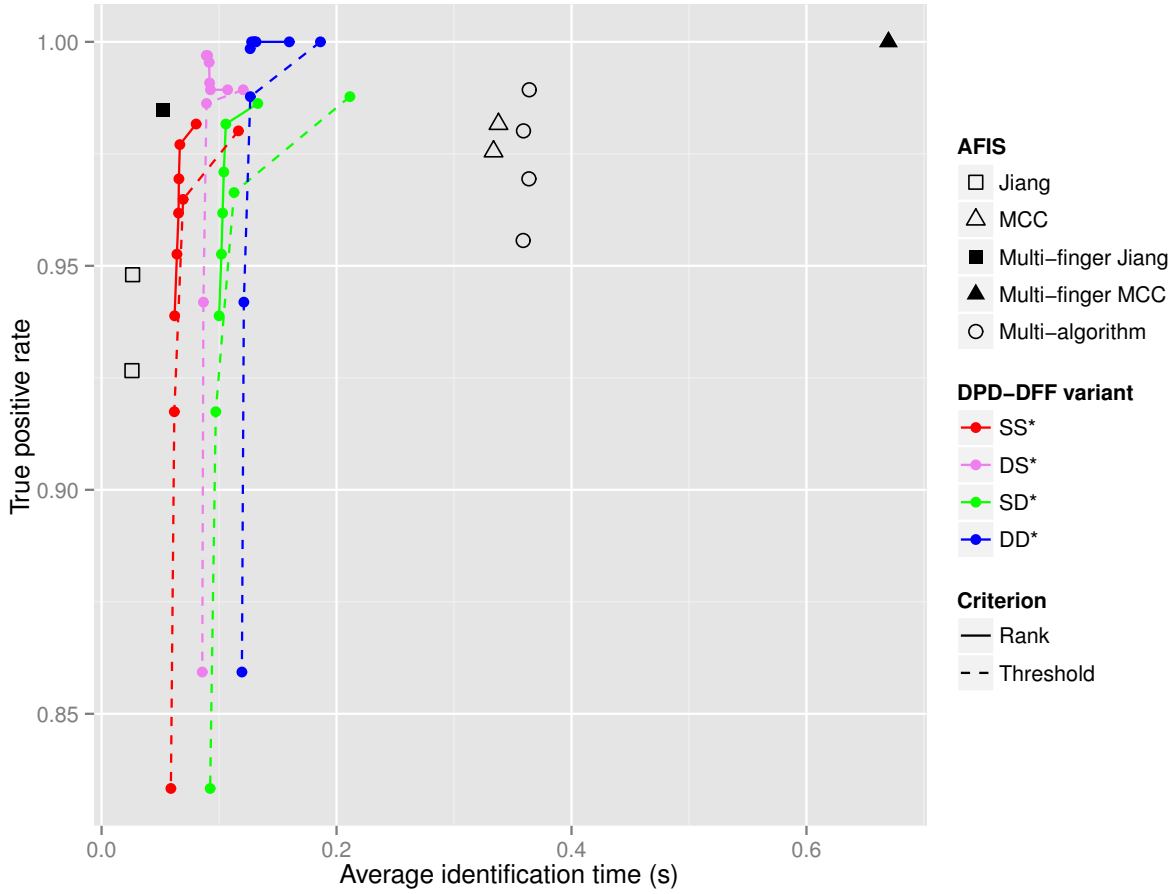


Figure 3: Runtime and accuracy with the captured database ($N = 654$)

- **FAR100, FAR1000**: FRR when FAR is 1% and 0.1%, respectively.
- **True Negatives (TN)**: number of input fingerprints that are not in the database, and are correctly detected as such.
- **True Negative Rate (TNR)**: quotient of TN and the number of impostor test fingerprints.

Tables 8 and 9 depict these error measures for all tested variants of DPD-DFF. To calculate these values, we took 3 random input fingerprint pairs for each of the 654 templates, and matched them with all the templates, making a total of 1 283 148 matchings for each matcher and each finger.

These tables show that the error rates become very low when the candidate set is big enough, especially for the DDR variant. Additionally, the FAR100 and FAR1000 values are very similar in most cases, meaning that the FAR drops quickly while the FRR remains almost constant, stating the robustness of DPD-DFF. The DRR variant obtains a very low FAR1000 when $r = 48$, which corresponds to a system that is robust against attacks (0.1% FAR), while avoiding rejections of genuine identities (0.25% FRR).

Table 8: Additional error measures (in percentages) using DPD-DFF (rank)

	SSR			SDR			DSR			DDR		
r	EER	FAR100	FAR1000	EER	FAR100	FAR1000	EER	FAR100	FAR1000	EER	FAR100	FAR1000
2	6.2691	6.2691	6.3462	6.2691	6.2691	6.2691	1.3252	1.3252	1.4547	1.3252	1.3252	1.3252
4	5.3007	5.3007	5.4536	5.3007	5.3007	5.3007	0.9684	0.9684	1.1879	0.9684	0.9684	0.9684
8	4.4852	4.4907	4.6406	4.4852	4.4852	4.4852	0.8396	0.7875	1.2571	0.7645	0.7645	0.7645
12	3.8226	3.9012	4.0571	3.8226	3.8226	3.8226	0.7715	0.7344	1.1956	0.6116	0.6116	0.6116
24	2.7405	2.8525	3.2449	2.7013	2.7013	2.7077	0.5750	0.5607	1.0508	0.3568	0.3568	0.3818
48	2.0346	2.0897	2.5592	1.8858	1.8858	1.9217	0.6132	0.4497	1.2210	0.2039	0.2039	0.2519

Table 9: Additional error measures (in percentages) using DPD-DFF (threshold)

	SST			SDT			DST			DDT		
θ	EER	FAR100	FAR1000	EER	FAR100	FAR1000	EER	FAR100	FAR1000	EER	FAR100	FAR1000
0.15	1.9888	2.0829	2.7641	1.7848	1.7848	1.8358	0.6938	0.5347	1.3761	0.3058	0.3058	0.3851
0.20	4.5385	4.6810	4.9608	4.5385	4.5385	4.5385	1.6820	1.6820	1.8941	1.6820	1.6820	1.6820
0.25	9.5360	9.5360	9.6470	9.5360	9.5360	9.5360	5.8104	5.8104	5.8104	5.8104	5.8104	5.8104
0.30	18.2050	18.2050	18.2050	18.2050	18.2050	18.2050	15.4944	15.4944	15.4944	15.4944	15.4944	15.4944

To conclude this section, we performed a new test, for which half of the fingerprints were randomly removed from the database, so that 50% of the input fingerprints become impostors trying to break into the system. The criterion used by DPD-DFF to determine if a fingerprint does not belong to the database is a score threshold within the accurate phase: if the fingerprint selected as the most similar to the input does not reach that threshold, the input is considered to be an impostor. The threshold used for these tests was the one that gives 0.01% FAR for the standalone MCC algorithm.

Table 10: Results of DPD-DFF with impostors and 654 test identifications (rank)

	SSR				**SDR**				**DSR**				**DDR**			
r	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
2	303	327	0	24	309	324	3	18	321	326	1	6	327	325	2	0
4	305	327	0	22	311	324	3	16	321	326	1	6	327	325	2	0
8	308	327	0	19	314	324	3	13	321	326	1	6	327	325	2	0
12	310	327	0	17	316	323	4	11	321	325	2	6	327	323	4	0
24	313	326	1	14	319	322	5	8	321	322	5	6	327	321	6	0
48	317	324	3	10	323	320	7	4	321	320	7	6	327	320	7	0

The results presented in Tables 10 and 11 show that, in contrast to the behavior of the TP, the TN decreases as the candidate size grows. This happens because a smaller candidate set allows the impostors to be detected during the first phase, while a bigger set makes the system more vulnerable to such attacks. This behavior provides good flexibility for the system: it can focus either on rejecting impostors or avoiding false rejections by modifying the rank parameter.

All rank variants of DPD-DFF show very good accuracy results when detecting impostors while identifying genuine fingerprints, keeping both FP and FN very low. As an example, DDR obtains the best result with the smallest r, and therefore the fastest configuration, without any false negatives in all cases. Similarly, the SSR variant is the one

Table 11: Results of DPD-DFF with impostors and 654 test identifications (threshold)

	SST				SDT				DST				DDT			
θ	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
0.15	316	323	4	11	322	320	7	5	321	319	8	6	327	321	6	0
0.20	308	326	1	19	314	323	4	13	319	326	1	8	325	324	3	2
0.25	295	327	0	32	301	325	2	26	307	327	0	20	311	327	0	16
0.30	260	327	0	67	264	327	0	63	279	327	0	48	281	327	0	46

that is more robust against false positives, although this happens at the cost of a worse false negative rate.

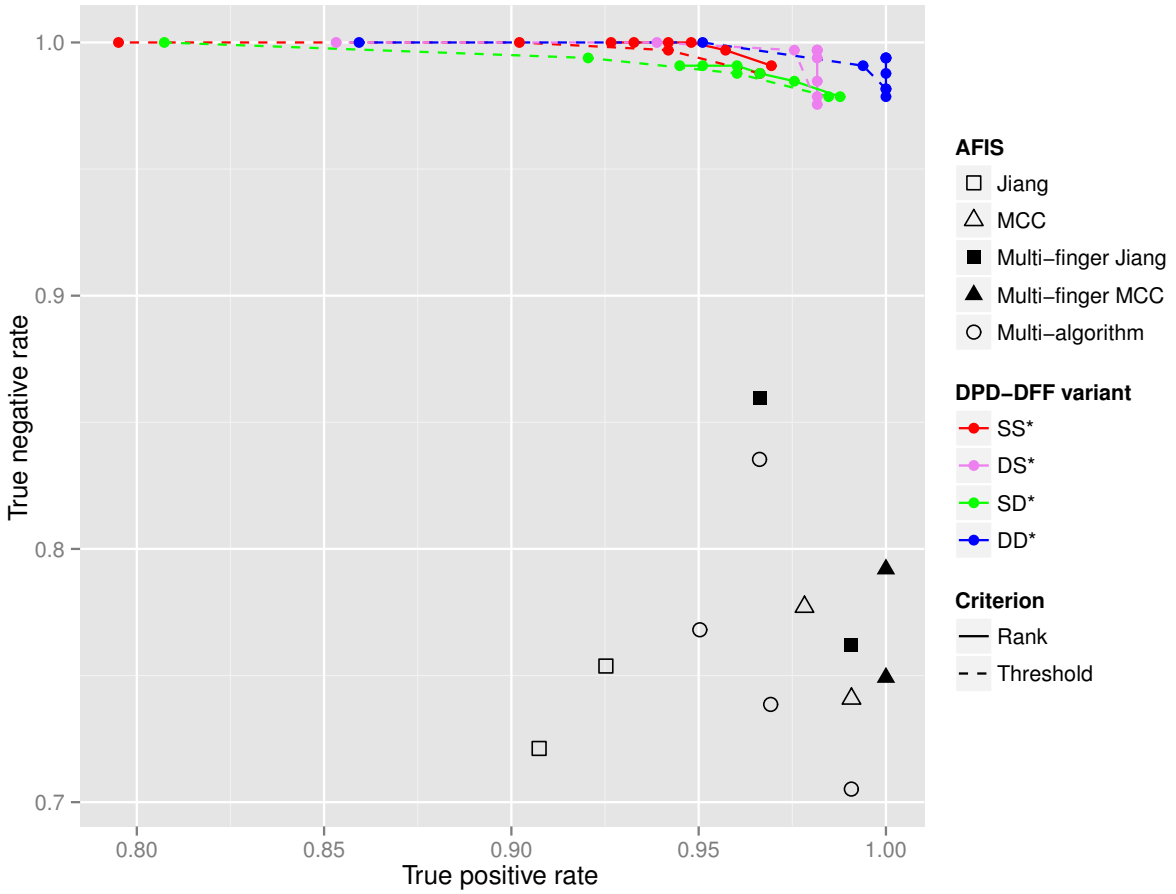


Figure 4: True positive rate and true negative rate with the captured database

Figure 4 shows how the proposed system dominates by far all reference AFIS in terms of the trade-off between false rejections and false acceptances. It can be seen that the DDR variant reaches almost 100% of both measures at the same time.

4.4. NIST-DB14 database

The NIST-DB14 database is composed of 27 000 rolled fingerprints, each of which was captured twice [44]. Tables 12 and 13 present the results of DPD-DFF over the NIST-DB14 database for the rank and threshold variants, respectively, using 12 slave machines. Figure 5 displays graphically the values of the tables.

Table 12: Results of DPD-DFF with 1000 test identifications (rank)

	SSR				**SDR**				**DSR**				**DDR**			
r	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)
12	244	756	0	2.0345	275	725	0	2.3242	335	665	0	3.5431	357	643	0	3.8281
24	259	741	0	2.0564	312	688	0	2.3920	357	643	0	3.5769	393	607	0	3.9234
48	272	728	0	2.0726	340	660	0	2.4209	367	633	0	3.5816	418	582	0	3.9301
96	287	713	0	2.0865	369	631	0	2.4353	382	618	0	3.5872	442	558	0	3.9446
144	294	706	0	2.0891	383	617	0	2.4418	388	612	0	3.5965	458	542	0	3.9531
288	303	697	0	2.1033	405	595	0	2.4644	393	607	0	3.6070	485	515	0	3.9681
576	311	689	0	2.2739	431	569	0	2.8361	391	609	0	3.7861	507	493	0	4.3407
1152	324	676	0	2.5856	458	542	0	3.5211	393	607	0	4.1169	528	472	0	5.0399
2304	318	682	0	3.1847	482	518	0	4.8292	395	605	0	4.7404	544	456	0	6.3622
4608	331	669	0	4.3706	499	501	0	7.3712	381	619	0	5.9571	556	444	0	8.9160
9216	336	664	0	6.6912	518	482	0	12.3243	369	631	0	8.2953	559	441	0	13.8824

Table 13: Results of DPD-DFF with 1000 test identifications (threshold)

	SST					**SDT**				**DST**					**DDT**			
ϕ	$\lvert C\rvert_{avg}$	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)	$\lvert C\rvert_{avg}$	TP	FP	FN	t_{avg} (s)	TP	FP	FN	t_{avg} (s)
0.10	13065.3	344	656	0	8.9011	535	465	0	17.0552	13380.5	360	640	0	10.7978	555	445	0	19.1454
0.15	3820.7	329	670	1	4.2518	484	515	1	7.2158	2197.2	394	605	1	4.9861	529	470	1	6.8838
0.20	367.3	298	684	18	2.2591	389	593	18	2.8427	49.7	342	540	118	3.5687	380	502	118	3.9022
0.25	6.0	191	476	333	1.9899	202	465	333	2.2579	0.2	137	31	832	3.4112	137	31	832	3.5791

It has to be noted that the TPR is surprisingly low, in discrepancy with other studies that highlight these matchers as accurate for the NIST-DB14 database. However, they may require specific tuning to be optimized for rolled fingerprints, which falls beyond the scope of this study. Therefore, we focus on the results obtained with general-purpose parameters that can highlight the robustness of the tested AFIS. For this database, which is difficult and computationally expensive, DPD-DFF outperforms all other approaches, in terms of both identification time and accuracy. The DDR variant is able to obtain better accuracy than the multi-finger MCC in about 25% more of the time than that required by the latter.

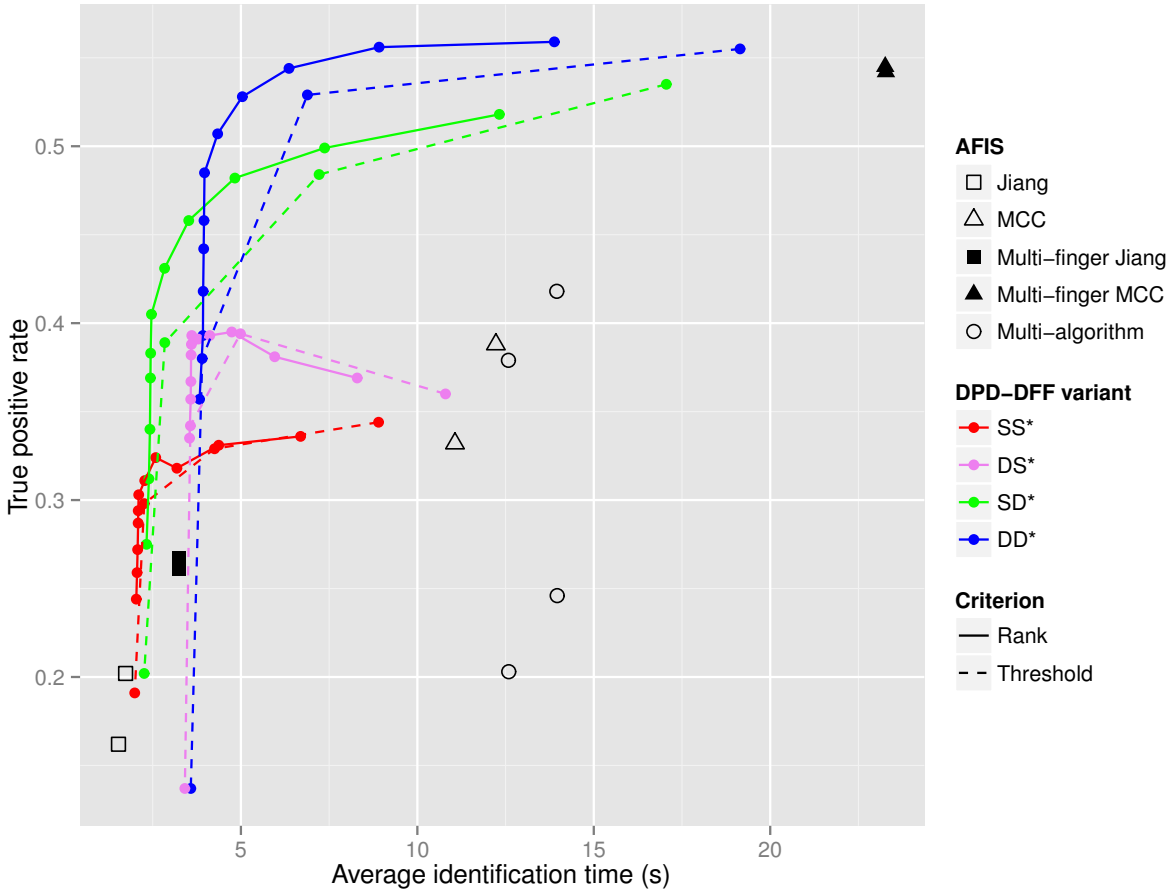


Figure 5: Average identification time and accuracy with NIST-DB14 (N = 21600)

4.5. Other real databases

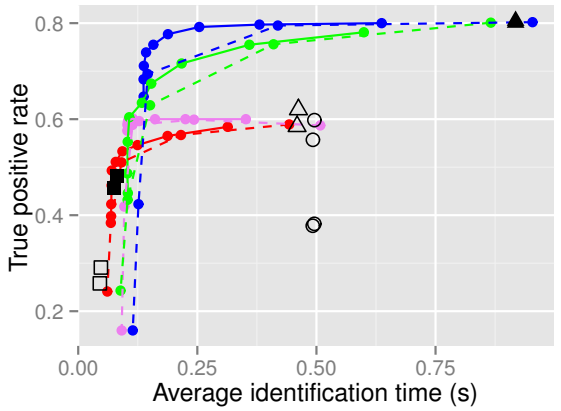
We have performed an extensive experimental study over several publicly available databases composed of real fingerprints. The objective of this study is to analyze the behavior of DPD-DFF in several realistic systems, where the fingerprints have been captured by different sensors and techniques, and to do so in a reproducible way. DB25496 is a mixture of the other four real databases formed by plain fingerprints (DBSpain654, CASIA-FingerprintV5, MCYT100 and FingerPass), where four captures of each fingerprint pair were included into the template database. Table 14 summarizes the characteristics of the three selected databases. Figure 6 displays graphically the time and accuracy values obtained for them.

Again, the DDR variant is able to obtain the same results as the multi-finger MCC, in a much shorter time frame. For the smaller databases, the multi-finger Jiang AFIS is able to obtain results that are faster than any of the variants of DPD-DFF. However, this time difference is less than 0.1s, which is usually an acceptable time to spend if the accuracy is improved. For a bigger database with the same characteristics, the relative difference in time would decrease as the accurate phase overhead would represent a smaller proportion of the overall time, thus making DPD-DFF even more

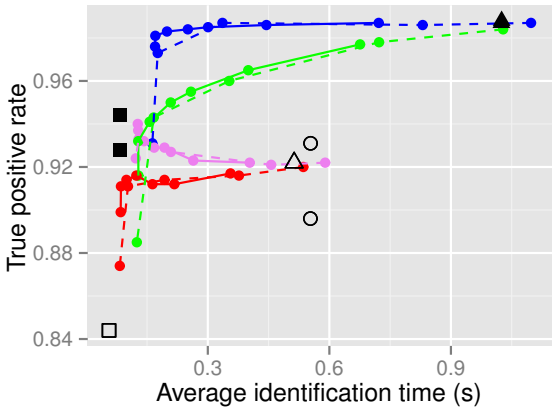
Table 14: Summary of the used real databases

Database	Subjects	Fingers	Template pairs	Input pairs	Machines used	Reference
CASIA-FingerprintV5	500	8	4000	1000	4	[45]
MCYT100	100	10	1000	1000	1	[46]
FingerPass	90	8	720	720	1	[47]
DB25496	1024	–	25 496	1000	12	[28]

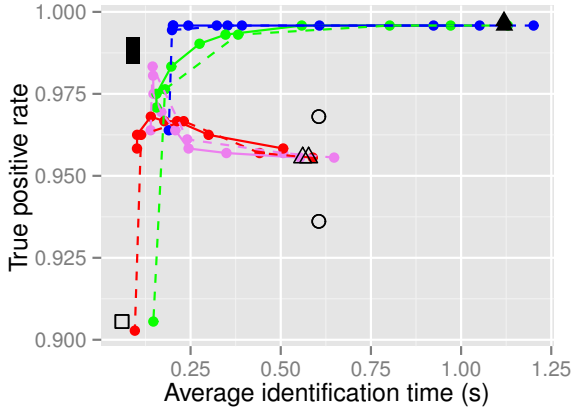
suitable for the identification. This is reflected in the largest and most difficult databases (CASIA-FingerprintV5 and DB25496), as well as in the previously analyzed NIST-DB14, where DPD-DFF improves accuracy in all the reference results.



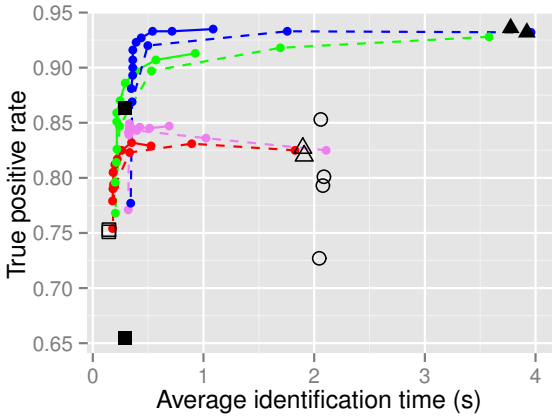
(a) CASIA-FingerprintV5



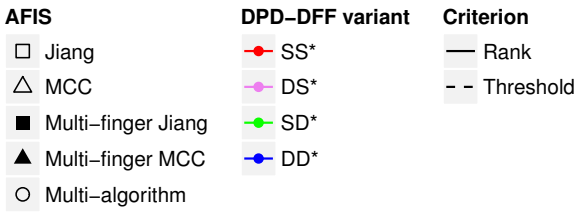
(b) MCYT100



(c) FingerPass



(d) DB25496



(e) Legend

Figure 6: Average identification time and accuracy with the additional real databases

5. Conclusions

In this paper, we have proposed a novel dual phase identification model (denoted DPD-DFF) to address the identification problem in large fingerprint databases. Its goal is to overcome the two problems that arise when dealing with this kind of database: the accuracy loss and the long runtime. To do so, the model combines two matching algorithms and two fingerprints per identity, using a mixed decision-level and score-level fusion, and has been implemented in a distributed system.

One of the main strengths of the proposed system is its flexibility, so that it can be tuned to the desired balance between accuracy and speed. Furthermore, the proposal has been tested over six fingerprint databases of diverse characteristics. The attained results have shown that the solutions obtained by our model dominate both in time and in accuracy over those obtained by using a single fingerprint or score fusion with either two fingerprints or two matchers, especially when large or complex databases are involved.

With a database of 50 000 fingerprint pairs, the algorithm reaches 100% TPR for identification taking only 0.44 seconds in a cluster of 12 machines. As for the fast results, 98.0% accuracy is obtained within 0.17 seconds.

The experiments carried out over the remaining databases have confirmed these conclusions. The additional study including impostor scores claims that DPD-DFF is much more precise than the three reference AFIS in terms of the trade-off between TPR and TNR, being able to eliminate any false negatives within a fast identification time.

Acknowledgments

This work was supported by the research projects TIN2014-57251-P, TIN2013-47210-P and P12-TIC-2958. D. Peralta holds an FPU scholarship from the Spanish Ministry of Education and Science (FPU12/04902). I. Triguero holds a BOF postdoctoral fellowship from the Ghent University.

Portions of the research in this paper use the CASIA-FingerprintV5 collected by the Chinese Academy of Sciences' Institute of Automation (CASIA).

References

[1] A. K. Jain, R. M. Bolle, S. Pankanti, Biometrics: Personal Identification in Networked Society, Springer, 2005.

[2] S. Pankanti, S. Prabhakar, A. K. Jain, On the individuality of fingerprints, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1010–1025.

[3] D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, Handbook of fingerprint recognition, Springer-Verlag New York Inc, 2009.

[4] A. Jain, L. Hong, R. Bolle, On-line fingerprint verification, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 302–314.

[5] A. K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity-authentication system using fingerprints, Proc. IEEE 85 (9) (1997) 1365–1388.

[6] N. Ratha, R. Bolle, Automatic Fingerprint Recognition Systems, Springer, New York, 2004.

[7] D. Peralta, M. Galar, I. Triguero, D. Paternain, S. García, E. Barrenechea, J. M. Benitez, H. Bustince, F. Herrera, A Survey on Fingerprint Minutiae-based Local Matching for Verification and Identification: Taxonomy and Experimental Evaluation, Inf. Sci. 315 (2015) 67–87.

[8] X. Jiang, W. Y. Yau, Fingerprint minutiae matching based on the local and global structures, in: Proc. 15th Int. Conf. Pattern Recognit., Vol. 2, IEEE, 2000, pp. 1038–1041.

[9] R. Cappelli, M. Ferrara, D. Maltoni, Minutia cylinder-code: A new representation and matching technique for fingerprint recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2128–2141.

[10] P. Verlinde, G. Chollet, M. Acheroy, Multi-modal identity verification using expert fusion, Inf. Fusion 1 (1) (2000) 17–33.

[11] S. Prabhakar, A. K. Jain, Decision-level fusion in fingerprint verification, Pattern Recognit. 35 (4) (2002) 861–874.

[12] A. Ross, A. Jain, Information fusion in biometrics, Pattern Recognit. Lett. 24 (13) (2003) 2115–2125.

[13] A. K. Jain, P. Flynn, A. A. Ross, Handbook of biometrics, Springer, 2007.

[14] A. K. Jain, S. Prabhakar, S. Chen, Combining multiple matchers for a high security fingerprint verification system, Pattern Recognit. Lett. 20 (1999) 1371–1379.

[15] X. Jiang, W. Ser, Online fingerprint template improvement, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1121–1126.

[16] H. Xu, R. N. J. Veldhuis, Spectral minutiae representations for fingerprint recognition, in: Proc. 6th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process., 2010, pp. 341–345.

[17] G. L. Marcialis, F. Roli, L. Didaci, Multimodal fingerprint verification by score-level fusion: An experimental investigation, J. Intell. Fuzzy Syst. 24 (2013) 51–60.

[18] N. K. Ratha, K. Karu, S. Chen, A. K. Jain, A real-time matching system for large fingerprint databases, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 799–813.

[19] H. S. Stone, High-performance computer architecture, Addison-Wesley Longman Publishing Co., Inc., 1992.

[20] R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, B. Smolinski, Toward a Common Component Architecture for High-Performance Scientific Computing, in: Proc. 8th IEEE Int. Symp. High Perform. Distrib. Comput., 1999, pp. 115–124.

[21] R. F. Miron, T. S. Letia, M. Hulea, Two server topologies for a distributed fingerprint-based recognition system, in: 15th Int. Conf. Syst. Theory, Control Comput., 2011, pp. 1–6.

[22] K. Beghdad Bey, Z. Guessoum, A. Mokhtari, F. Benhammadi, Agent based approach for distribution of fingerprint matching in a metacomputing environment, in: Proc. 8th Int. Conf. New Technol. Distrib. Syst., 2008, pp. 1–7.

[23] D. Peralta, I. Triguero, R. Sanchez-Reillo, F. Herrera, J. M. Benitez, Fast Fingerprint Identification for Large Databases, Pattern Recognit. 47 (2) (2014) 588–602.

[24] P. D. Gutierrez, M. Lastra, F. Herrera, J. M. Benitez, A high performance fingerprint matching system for large databases based on GPU, IEEE Trans. Inf. Forensics Secur. 9 (1) (2014) 62–71.

[25] R. Cappelli, M. Ferrara, D. Maltoni, Large-scale fingerprint identification on GPU, Inf. Sci. 306 (2015) 1–20.

[26] Unique Authentication Authority of India.
URL `http://uidai.gov.in/`

[27] Integrated Automated Fingerprint Identification System.
URL `http://www.fbi.gov/about-us/cjis/fingerprints\_biometrics/iafis/iafis`

[28] D. Peralta, I. Triguero, S. García, F. Herrera, J. M. Benitez, Supplementary material for "DPD-DFF: A Dual Phase Distributed Scheme with Double Fingerprint Fusion for Fast and Accurate Identification in Large Databases", Tech. rep., Soft Computing and Intelligent Information Systems, University of Granada (2015).
URL `http://sci2s.ugr.es/sites/default/files/files/ComplementaryMaterial/DPDDFF/techrep.pdf`

[29] D. Peralta, M. Galar, I. Triguero, J. M. Benitez, F. Herrera, Minutiae Filtering to Improve Both Efficacy and Efficiency of Fingerprint Matching Algorithms, Eng. Appl. Artif. Intell. 32 (2014) 37–53.

[30] A. Lindoso, L. Entrena, J. Izquierdo, FPGA-based acceleration of fingerprint minutiae matching, in: Proc. 3rd South. Conf. Program. Log., 2007, pp. 81–86.

[31] X. Jiang, M. Liu, A. C. Kot, Fingerprint retrieval for identification, IEEE Trans. Inf. Forensics Secur. 1 (4) (2006) 532–542.

[32] M. Liu, X. Jiang, A. Chichung Kot, Efficient fingerprint search based on database clustering, Pattern Recognit. 40 (6) (2007) 1793–1803.

[33] R. Cappelli, M. Ferrara, D. Maltoni, Fingerprint indexing based on minutia cylinder-code, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 1051–1057.

[34] M. Galar, J. Derrac, D. Peralta, I. Triguero, D. Paternain, C. Lopez-Molina, S. García, J. M. Benitez, M. Pagola, E. Barrenechea, H. Bustince, F. Herrera, A survey of fingerprint classification Part I: Taxonomies on feature extraction methods and learning models, Knowledge-Based Syst. 81 (2015) 76–97.

[35] M. Vatsa, R. Singh, A. Noore, K. Morris, Simultaneous latent fingerprint recognition, Appl. Soft Comput. 11 (7) (2011) 4260–4266.

[36] L. Nanni, D. Maio, Combination of different fingerprint systems: A case study FVC2004, Sens. Rev. 26 (1) (2006) 51–57.

[37] A. Noore, R. Singh, M. Vatsa, Robust memory-efficient data level information fusion of multi-modal biometric images, Inf. Fusion 8 (4) (2007) 337–346.

[38] T. Uz, G. Bebis, A. Erol, S. Prabhakar, Minutiae-based template synthesis and matching for fingerprint authentication, Comput. Vis. Image Underst. 113 (2009) 979–992.

[39] Y. Li, J. Yin, E. Zhu, Score-based fusion in multi-unit biometric recognition system, Appl. Mech. Mater. 48–49 (2011) 1010–1013.

[40] D. Gafurov, C. Busch, P. Bours, B. Yang, Fusion in fingerprint authentication: Two finger types vs. two scanner types, in: Proc. ACM Symp. Appl. Comput., 2011, pp. 13–20.

[41] A. K. Jain, S. Prabhakar, A. Ross, Fingerprint matching: Data acquisition and performance evaluation, Tech. Rep. TR99-14, MSU (1999).

[42] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. Mccabe, S. Janet, K. Ko, User's Guide to NIST Biometric Image Software (NBIS), Tech. Rep. NISTIR-7392, NIST (2010).

[43] R. Cappelli, D. Maio, D. Maltoni, Synthetic fingerprint-database generation, in: Proc. 16th Int. Conf. Pattern Recognit., Vol. 3, 2002, pp. 744–747.

[44] C. I. Watson, NIST Special Database 14, Tech. rep., NIST (1993).

[45] CASIA-FingerprintV5 database.
URL `http://biometrics.idealtest.org/`

[46] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero, Q.-I. Moro, MCYT baseline corpus: A bimodal biometric database, IEEE Proc. Vision, Image Signal Process. 150 (6) (2003) 395–401.

[47] X. Jia, X. Yang, Y. Zang, N. Zhang, J. Tian, A cross-device matching fingerprint database from multi-type sensors, in: Proc. Int. Conf. Pattern Recognit., 2012, pp. 3001–3004.